

BFS

최백준 choi@startlink.io

구슬 탈출 4

<https://www.acmicpc.net/problem/15653>

- 보드의 상태가 주어졌을 때, 최소 몇 번 만에 빨간 구슬을 구멍을 통해 빼낼 수 있는지 구하는 문제
- 만약, 어떻게 움직여도 빨간 구슬을 구멍을 통해 빼낼 수 없으면 -1을 출력

구슬 탈출 4

<https://www.acmicpc.net/problem/15653>

- 보드의 상태를 변수로 나타내보자

구슬 탈출 4

<https://www.acmicpc.net/problem/15653>

- 보드에는 빈 칸, 벽, 빨간 구슬, 파란 구슬, 구멍이 있다.
- 가능한 방법은 구슬을 이동시키는 것이다.

구슬 탈출 4

<https://www.acmicpc.net/problem/15653>

- 보드에는 빈 칸, 벽, 빨간 구슬, 파란 구슬, 구멍이 있다.
- 가능한 방법은 구슬을 이동시키는 것이다.
- 빈 칸, 벽, 구멍은 어떻게 이동시켜도 변하지 않는다.
- 빨간 구슬과 파란 구슬만 변한다.
- 따라서, 구슬의 위치가 문제의 상태가 된다.

구슬 탈출 4

<https://www.acmicpc.net/problem/15653>

- 상태: (rx, ry, bx, by)
- 빨간 구슬의 위치가 (rx, ry)이고, 파란 구슬의 위치가 (bx, by)
- 총 가능한 상태의 개수: $(NM)^2$

구슬 탈출 4

7

<https://www.acmicpc.net/problem/15653>

- 소스: <http://codeplus.codes/95632f244cbf45949a6833419a05c24e>

과외맨

<https://www.acmicpc.net/problem/5213>

- 홀수 줄에는 N개의 타일, 짝수 줄에는 N-1개의 타일이 있다
- 첫 줄의 첫 타일에서 마지막 줄의 마지막 타일로 가는 가장 빠른 시간을 구하는 문제
- 한 타일에서 다른 타일로 이동하려면, 인접해야 하고, 변을 공유하는 조각에 써 있는 수가 같아야 한다

1	4	4	5	3	4	5	4	5	2
	4	2	5	6	4	4	6	5	
2	4	5	1	6	1	1	6	2	3
	4	2	5	3	1	2	5	5	
4	1	2	2	4	3	2	3	3	4

과외맨

<https://www.acmicpc.net/problem/5213>

- 소스: <http://codeplus.codes/b449fdb15a9c463ab79e7869f7404270>

양

<https://www.acmicpc.net/problem/3184>

- 지도는 영역으로 나누어져 있고
- 각 영역에서 양의 수와 늑대의 수를 구한다
- 양 > 늑대 -> 양
- 양 ≤ 늑대 -> 늑대

양

<https://www.acmicpc.net/problem/3184>

- 영역을 구하면서 극대와 양의 수를 구할 수 있다

<https://www.acmicpc.net/problem/3184>

- 영역은 DFS/BFS로 구할 수 있기 때문에, 이 방법을 사용한다

<https://www.acmicpc.net/problem/3184>

- 소스: <http://codeplus.codes/16b06a0efc094aa48c72f8f60cced339>

스타트링크

<https://www.acmicpc.net/problem/5014>

- 스타트링크는 총 F층으로 이루어진 건물에 사무실이 있다. 사무실은 G층
- 지금 S층에 있는 사람이 G층으로 가려면 버튼을 최소 몇 번 눌러야 하는지 구하는 문제
- 엘리베이터의 버튼은 U (U층 위로 올라가는 버튼), D (D층 아래로 내려가는 버튼)
- $1 \leq S, G \leq F \leq 1000000, 0 \leq U, D \leq 1000000$

스타트링크

<https://www.acmicpc.net/problem/5014>

- 총 백만개의 층으로 이루어져 있고, 여기서 BFS탐색을 이용하면 최소로 버튼을 누르는 횟수를 구할 수있다.

스타트링크

<https://www.acmicpc.net/problem/5014>

- 소스: <http://codeplus.codes/05155fa1d0df442998e3cf1007daa0ea>

돌 그룹

<https://www.acmicpc.net/problem/12886>

- 돌이 3개의 그룹으로 나누어져 있고, 각 그룹에는 돌이 A, B, C개 있다. ($A, B, C \leq 500$)
- 돌은 단계별로 움직이고, 각 단계는 다음과 같다.
- 크기가 같지 않은 두 그룹을 고른다. 돌의 개수가 작은 쪽을 X, 큰 쪽을 Y라고 한다.
- X에 있는 돌의 개수를 $X+X$ 개로, Y에 있는 돌의 개수를 $Y-X$ 로 만든다.
- A, B, C가 주어졌을 때, 모든 그룹에 들어있는 돌의 개수를 같게 만들 수 있는지 구하는 문제

돌 그룹

<https://www.acmicpc.net/problem/12886>

- BFS로 해결할 수 있다.
- 정점: (A, B, C) 또는 (A, B)
- 전체 정점의 개수: $A+B+C$ 개

돌 그룹

<https://www.acmicpc.net/problem/12886>

- 소스: <http://codeplus.codes/2b591df0257b40b9898f80dc1eb58869>

벽 부수고 이동하기 2

<https://www.acmicpc.net/problem/14442>

- $N \times M$ 의 행렬로 나타내는 지도에서 (1, 1)에서 (N,M)으로 최단 거리로 이동하는 문제
- 0은 빈 칸, 1은 벽
- 단, 벽은 K번까지 부수고 지나갈 수 있다

벽 부수고 이동하기 2

<https://www.acmicpc.net/problem/14442>

- 벽을 부순다는 조건이 없으면 일반적인 미로 탐색 문제이다
- 어떤 칸에 방문했을 때, 벽을 부순 적이 있는 경우와 아직 부순 적이 없는 경우는 다른 경우이기 때문에
- 상태 (i, j) 대신에 (i, j, b) (b 는 벽을 부순 횟수) 으로 BFS 탐색을 진행한다.

벽 부수고 이동하기 2

22

<https://www.acmicpc.net/problem/14442>

- 소스: <http://codeplus.codes/a56bc7af79114efe9da7e935a4d3cb13>

말이 되고픈 원숭이

<https://www.acmicpc.net/problem/1600>

- 격자판 위에 원숭이가 있다.
- 원숭이의 시작 칸에서 도착 칸으로 가는 최소 이동 동작의 횟수를 구하는 문제
- 원숭이는 인접한 4방향만 이동할 수 있고, 최대 K번 체스판의 나이트처럼 이동할 수 있다.

말이 되고픈 원숭이

<https://www.acmicpc.net/problem/1600>

- 벽 부수고 이동하기 2 문제와 같이 나이트처럼 이동을 한 횟수에 따라서, 정점을 나누어서 BFS를 구현할 수 있다.

말이 되고픈 원숭이

<https://www.acmicpc.net/problem/1600>

- 격자판 위에 원숭이가 있다.
- 원숭이의 시작 칸에서 도착 칸으로 가는 최소 이동 동작의 횟수를 구하는 문제
- 원숭이는 인접한 4방향만 이동할 수 있고, 최대 K번 체스판의 나이트처럼 이동할 수 있다.

말이 되고픈 원숭이

26

<https://www.acmicpc.net/problem/1600>

- 소스: <http://codeplus.codes/0cb43fbb48b14b23a3b59732814b89c3>

적록색약

<https://www.acmicpc.net/problem/10026>

- $N \times N$ 크기의 격자가 있고, 각 칸에는 R, G, B 중 하나의 색이 색칠되어져 있다.
- 같은 색상이 인접하는 경우 두 구역은 같은 그림이다.
- 적록색약인 사람은 빨간색과 초록색의 차이를 느끼지 못한다.

RRRBB

GGBBB

BBBRR

BBRRR

RRRRR

- 적록색약인 사람이 보면 구역: 3개, 아닌 사람이 보면 4개
- $1 \leq N \leq 100$

적록색약

<https://www.acmicpc.net/problem/10026>

- BFS를 일반 사람이 봤을 때와 적록색약이 봤을 때로 나누어서 두 번 구현하면 된다.

적록색약

<https://www.acmicpc.net/problem/10026>

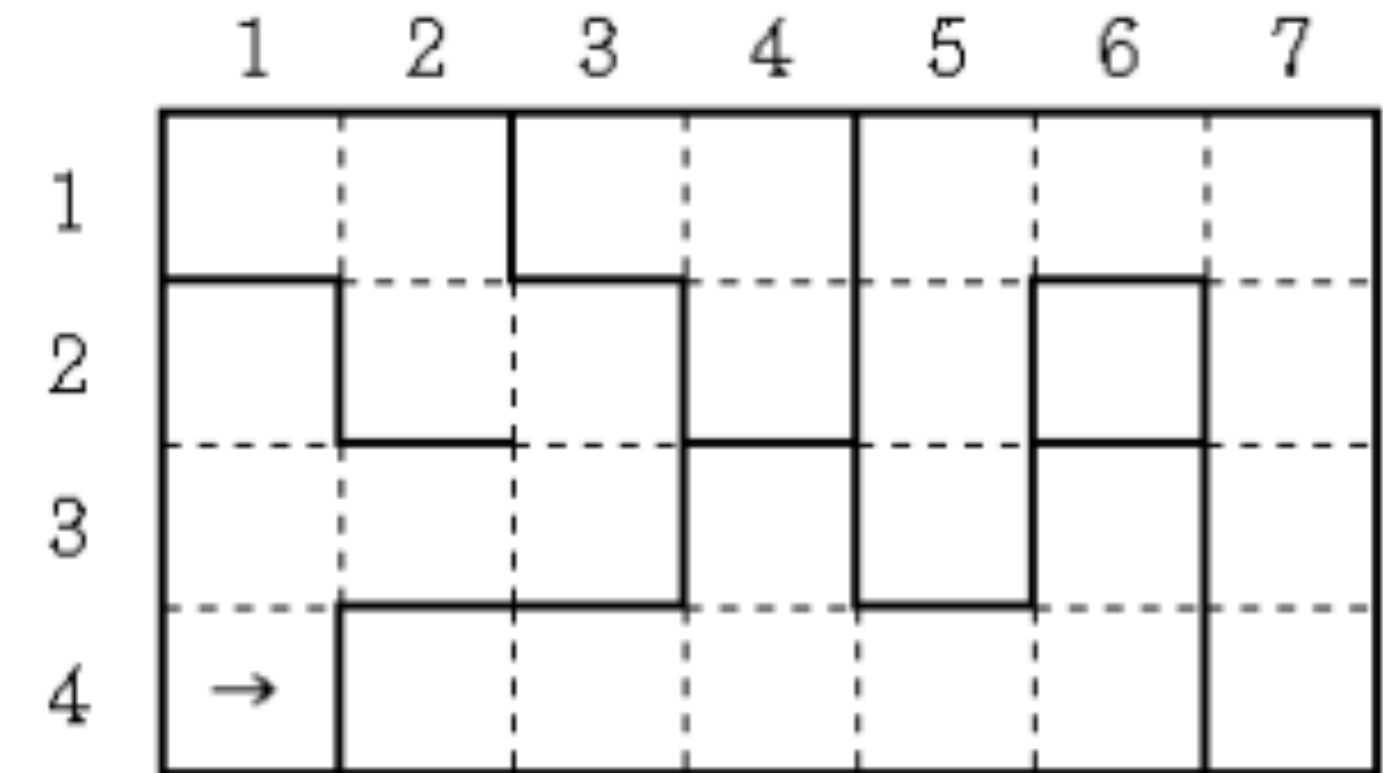
- 소스: <http://codeplus.codes/f90f68470abe4519abe747645713a614>

성곽

30

<https://www.acmicpc.net/problem/2234>

- 성곽이 있다
- 1. 방의 개수
- 2. 가장 넓은 방의 넓이
- 3. 하나의 벽을 제거해서 얻을 수 있는 가장 넓은 방의 크기

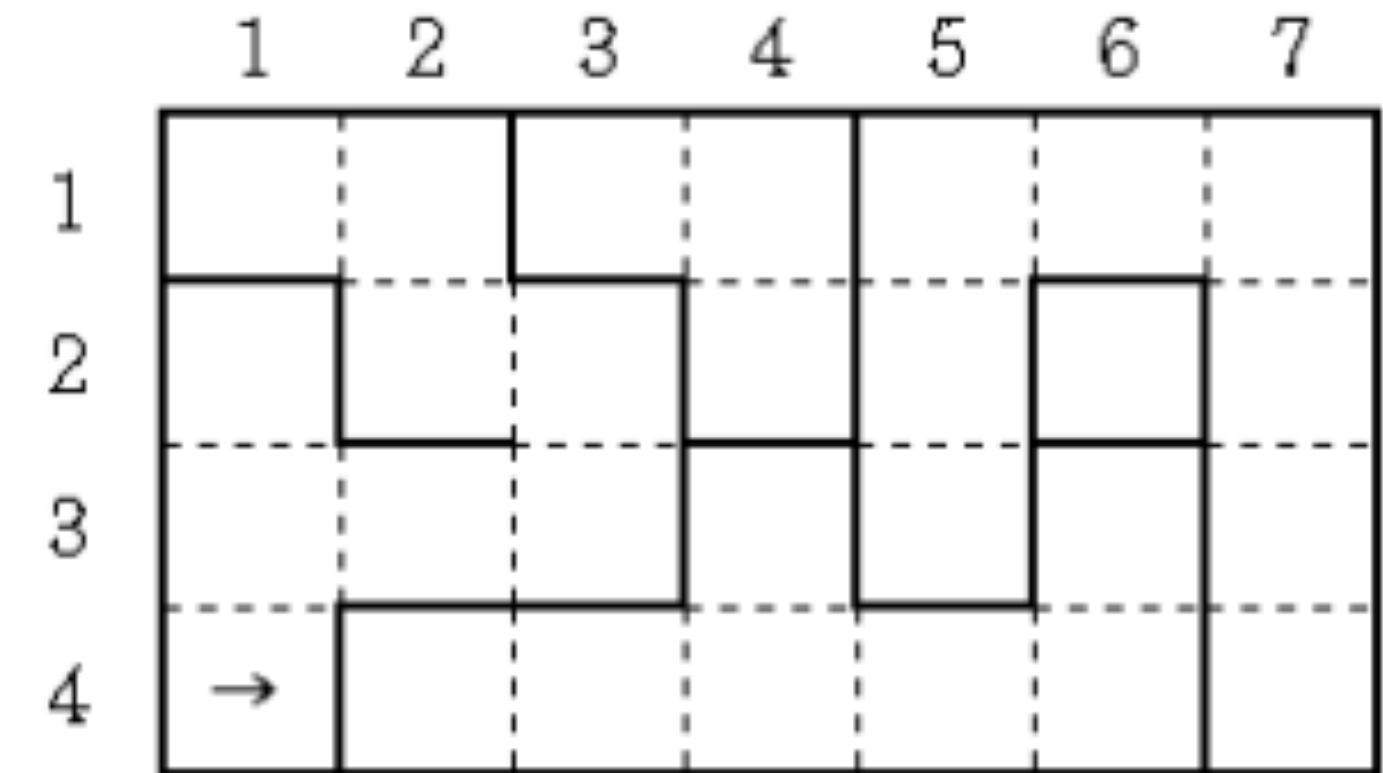


성곽

31

<https://www.acmicpc.net/problem/2234>

- 1, 2번 문제를 풀기 위해 BFS를 이용해서
- 각 칸의 방 번호와
- 각 방의 크기를 모두 구할 수 있다

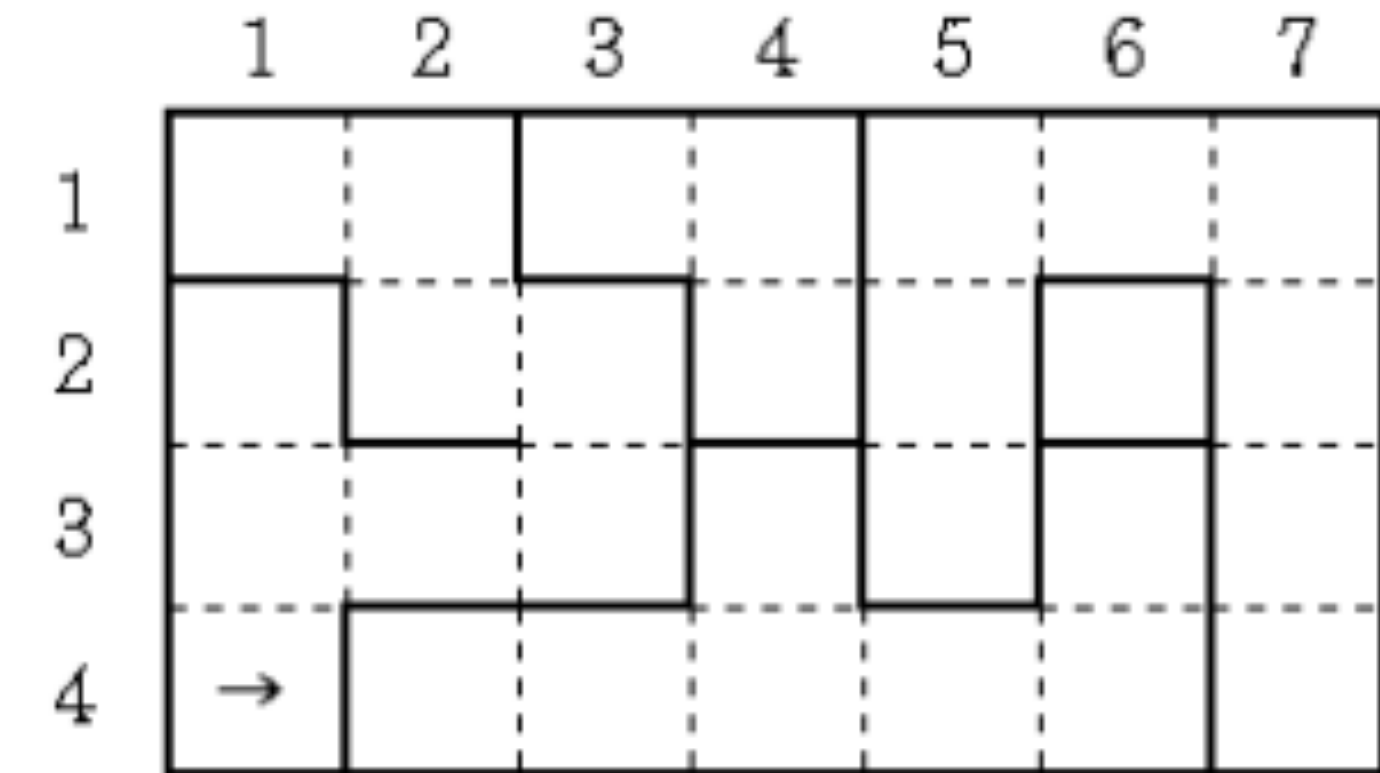


성곽

32

<https://www.acmicpc.net/problem/2234>

- 3번 문제를 풀기 위해 각각의 벽을 하나씩 없애보고
- BFS를 돌리는 것은 너무 오래 시간이 걸린다
- 1, 2번 문제를 풀면서 얻은 정보를 이용해
- 모든 벽에 대해서, 각 벽을 제거했을 때 방이 합쳐지는지
- 그리고, 넓이는 몇인지 구할 수 있다



성곽

<https://www.acmicpc.net/problem/2234>

- 소스: <http://codeplus.codes/fb94b78f0029411a8a9bba03645bd96f>

백조의 호수

<https://www.acmicpc.net/problem/3197>

- 두 마리의 백조가 호수에 살고 있다
- 두 마리는 호수를 덮고 있는 얼음 때문에 만날 수 없다
- 매일 물과 접촉한 얼음은 녹는다 (가로, 세로)
- 몇 일이 지나야 백조가 만날 수 있을까?

백조의 호수

<https://www.acmicpc.net/problem/3197>

```
...XXXXXX..XX.XXX  ...XXXX.....XX  ...XX.....
...XXXXXXXXXX.XXX  ...XXXX..X.....  ...X.....
...XXXXXXXXXXXXX..  ...XXX..XXXX....  ...X...X....
..XXXXX.LXXXXXX..  ...XXX...XXXX....  ...X.....XX....
.XXXXXX..XXXXXX..  ..XXXX...XXXX....  ...XX.....XX....
XXXXXXXXX...XXXX...  ..XXXX.....XX....  ...X.....
..XXXXX...XXX....  ...XX...X.....
...XXXXX.XXXL...  ...XX...X.....
```

시작

첫째 날

둘째 날

백조의 호수

<https://www.acmicpc.net/problem/3197>

- 물의 퍼짐과 백조의 이동을 BFS로 진행할 수 있다
- 각각의 날에 대해서
- 물을 먼저 이동시키고, 그 다음에 백조를 이동시키면 된다

백조의 호수

<https://www.acmicpc.net/problem/3197>

- 소스: <http://codeplus.codes/310280afb0aa4400aaad69681e9e5e8c>

새로운 하노이 탑

<https://www.acmicpc.net/problem/12906>

- 총 3가지 종류의 막대가 있다. 막대 A, B, C
- 각 막대에는 원판이 0개 이상 놓여져 있다.
- 모든 원판의 크기는 같고, 원판의 종류도 A, B, C이다.
- 한 번 이동: 한 막대의 가장 위에 있는 원판을 다른 막대의 가장 위로 옮기는 것
- 막대 A에는 원판 A만, 막대 B에는 원판 B만, 막대 C에는 원판 C만 있게 옮겨야 한다.
- 최소 이동 횟수를 구하는 문제
- $1 \leq$ 모든 막대에 놓여져 있는 원판 개수의 합 ≤ 10

새로운 하노이 탑

39

<https://www.acmicpc.net/problem/12906>

1 B

1 C

1 A

- 원판 A를 막대 A로
- 원판 C를 막대 C로
- 원판 A를 막대 C로
- 원판 B를 막대 B로
- 원판 A를 막대 A로

새로운 하노이 탑

40

<https://www.acmicpc.net/problem/12906>

3 CBA

0

0

- 원판 A를 막대 C로
- 원판 B를 막대 B로
- 원판 A를 막대 B로
- 원판 C를 막대 C로
- 원판 A를 막대 A로

새로운 하노이 탑

<https://www.acmicpc.net/problem/12906>

- 막대의 개수가 3개, 원판의 총 개수가 10개밖에 안된다.
- 경우의 수가 많지 않아 BFS 탐색으로 모든 이동을 해본다.

새로운 하노이 탑

42

<https://www.acmicpc.net/problem/12906>

- 소스: <http://codeplus.codes/124f2b5868594c69bc859999d133b883>

4 연산

<https://www.acmicpc.net/problem/14395>

- 정수 s 의 값을 t 로 바꾸는 최소 연산 횟수를 구하는 문제
- $s = s + s$; (출력: +)
- $s = s - s$; (출력: -)
- $s = s * s$; (출력: *)
- $s = s / s$; (출력: /) (s 가 0이 아닐때만 사용 가능)
- $1 \leq s, t \leq 10^9$

4 연산

<https://www.acmicpc.net/problem/14395>

- BFS 탐색 문제이다
- 만들어지는 수의 개수는 최대 10^9 개 이다!?

4 연산

<https://www.acmicpc.net/problem/14395>

- BFS 탐색 문제이다
- 만들어지는 수의 개수는 최대 10^9 개 이다!?
- 아니다. $x \rightarrow x^2$ 또는 $2 \cdot x$ 의 형태로 변형만 가능하기 때문에
- 만들어지는 수는 $x^a 2^b$ 의 형태이다.

4 연산

<https://www.acmicpc.net/problem/14395>

- 소스: <http://codeplus.codes/4760866c183a47fd993f91588d39c97b>

거울 설치

<https://www.acmicpc.net/problem/2151>

- 문이 두 개 있을 때, 한 문에서 다른 문을 보기 위해 필요한 거울 개수의 최소값을 구하는 문제

*	*	*	#	*
*	.	!	.	*
*	!	.	!	*
*	.	!	.	*
*	#	*	*	*

거울 설치

<https://www.acmicpc.net/problem/2151>

- 문이 두 개 있을 때, 한 문에서 다른 문을 보기 위해 필요한 거울 개수의 최소값을 구하는 문제

*	*	*	#	*
*	!	!	!	*
*	.	!	!	*
*	!	.	!	*
*	#	*	*	*

거울 설치

<https://www.acmicpc.net/problem/2151>

- 각각의 거울마다 네 방향에 대해서 다음 거울을 미리 찾아야 한다.

*	*	*	#	*
*	!	!	!	*
*	.	!	!	*
*	!	.	!	*
*	#	*	*	*

거울 설치

<https://www.acmicpc.net/problem/2151>

- 각각의 거울마다 네 방향에 대해서 다음 거울을 미리 찾아야 한다.
- 그 다음, 각각의 거울만 가지고 BFS 탐색을 수행해야 한다.

*	*	*	#	*
*	!	!	!	*
*	.	!	!	*
*	!	.	!	*
*	#	*	*	*

*	*	*	#	*
*	!	!	!	*
*	.	!	!	*
*	!	.	!	*
*	#	*	*	*

거울 설치

<https://www.acmicpc.net/problem/2151>

- 각각의 거울마다 네 방향에 대해서 다음 거울을 미리 모두 찾아야 한다.
- 그 다음, 각각의 거울만 가지고 BFS 탐색을 수행해야 한다.

*	*	*	#	*
*	!	!	!	*
*	.	!	!	*
*	!	.	!	*
*	#	*	*	*

*	*	*	#	*
*	!	!	!	*
*	.	!	!	*
*	!	.	!	*
*	#	*	*	*

거울 설치

52

<https://www.acmicpc.net/problem/2151>

- 소스: <http://codeplus.codes/2b0d9017322c4ea6ba54ae33ca4b9ccb>

견우와 직녀

<https://www.acmicpc.net/problem/16137>

- $N \times N$ 크기의 지도가 주어졌을 때, (0, 0)에서 (N-1, N-1)로 가는 가장 빠른 방법을 구하는 문제
- 오작교의 주기가 T이면 T초마다 오작교가 생긴다.
- 연속해서 오작교를 건널 수 없다.
- 절벽이 교차하는 경우 오작교를 만들 수 없다.
- 빈 칸을 하나만 오작교로 만들 수 있다. 주기 = M
- 0 : 절벽, 지나갈 수 없음
- 1: 땅, 지나갈 수 있음
- 2 이상: 오작교의 주기
- $N \leq 10$, 주기 ≤ 20

1	1	1	1	1
0	6	0	0	0
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1

견우와 직녀

<https://www.acmicpc.net/problem/16137>

- 오작교 때문에, 정점의 정보를 (r, c) 에서 (r, c, t) 와 같이 현재 이동한 시간을 포함해야 한다.
- 이 때, t 는 시간을 오작교의 주기로 나눈 값이다.
- 일단 오작교에 올라간 이동할 수 있을 때까지 오작교에 머물러 있는 것으로 처리한다.
- 이동할 수 있는 시간이 다른 칸으로 이동 가능, 아니면 현재 칸에 머무름

견우와 직녀

<https://www.acmicpc.net/problem/16137>

- 소스: <http://codeplus.codes/72e9ee70fd244507852fffe4b01ae109>

끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.